UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/662,095	09/12/2003	Ashish Thusoo	50277-2256	8871

42425 7590 11/15/2007

HICKMAN PALERMO TRUONG & BECKER/ORACLE
2055 GATEWAY PLACE
SUITE 550
SAN JOSE, CA 95110-1089

EXAMINER	
BELL, CORY C	

ART UNIT	PAPER NUMBER
2164	

MAIL DATE	DELIVERY MODE
11/15/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/662,095
Filing Date: September 12, 2003
Appellant(s): THUSOO, ASHISH

Daniel Ledesma
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 8/21/2007 appealing from the Office action mailed 12/22/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

PL/SQL User's Guide and Reference, Release 2 (9.2), Published March 2002, Chapter 5, pages 1. 51-55, Chapter 6 Pages 1-2, and Chapter 12 Pages 1-13. Known hereafter as [A]

Appellant's admitted prior art [AAPA] in the Background of appellant's specification conventional implementations of multi row aggregates using the RETURNING clause. Applicant has reproduced this section in pages 4-5 of the instant brief.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action: (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-71 10, and 15-20 are rejected under 35 U.S.C. 103 as being obvious over PL/SQL User's Guide and Reference Release 2 (9.2) hereafter know as [A] (This reference was included with the information discloser statement and has been labeled A) or [A] in view of the applicant admitted prior art[aapa].

Claim 1 is rejected for the following reasons: [A] Teaches, receiving a database statement that specifies a DML operation that modifies data in one or more columns in a database,(Page 53 of chapter 5, which teaches modifying a "sal" column, this is written in SQL which is a DML which

is inherently recieved) and contains a clause that specifies an aggregate operation to be performed on a plurality of values associated with the data(Page 53 of chapter 5, teaches aggregating that values of the ename, jab, and sal variables into an emp_info variable); and I in response to receiving the database statement, performing: the DML operation on the one or more columns in the database. performing the aggregate operation on the plurality of values(the query is inherently run and thus performing these features) and returning as a result of the database statement a result of the aggregate operation(Page 53 of chapter 5, teaches aggregating that values of the ename, jab, and sal variables into an emp_info variable which is then returned). [A] However fails to expressly disclose the use of an aggregate function for multiple rows in a returning clause. However, it would have been obvious to one of ordinary skill in the art to include aggregate functions as presently amended. As can be seen for the following reasons: First, the RETURNING clause is intended to eliminate the need for a select clause, [A] page 9 "This eliminates the need to SELECT the row after an insert or update, or before a delete. "Thus to truly eliminate the need for the SELECT clause it would need to integrate all the features of the SELECT clause, i.e. the ability to perform aggregate functions as claimed. Which are used to, as stated on pages 9 and 10 of reference [A], "As a result, fewer network round trips, less server CPU time, fewer cursers, and less server memory are required. "Thus, it would have been obvious to one of ordinary skill in that art to provide these feature in the return clause to provide the advantage of full eliminating the need for the select after an insert or update or before a delete. Reference [A] also shows the need to aggregate data further as it states "Now do computations involving name and new_sal. " The [aapa] teaches providing the desired agragate functions in the computations in para 11 of the background.

Claim 2 is rejected because the aggregation occurs when the change is prefonned as they are in the same database statement.

Claim 3 The method of claim 1, wherein the modified data includes values of the da.ta before the DML operation (sal inherently has a initial value else the update expression would be invalid).

Claim 4 is rejected because the data must inherently have a value that it is changed to when it is updated.

Claim 5 is rejected for reasons shown in the rejection of claim 1.

Claim 6 is rejected because the change being a deletion is shown in the last paragraph of page 9 of [A] section

Claim 7 is rejected because the method inherently has an SQL engine to process the SQL statements.

Claim 10 is rejected because the system inherently has a client interface to submit a database statement.

Claim 15 is rejected for the following reasons: See Claim 1 rejection.

Claim 16 is rejected for the following reasons: See Claim 2 rejection.

Claim 17 is rejected for the following reasons: See Claim 3 rejection.

Claim 18 is rejected for the following reasons: See Claim 4 rejection.

Claim 19 is rejected for the following reasons: See Claim 5 rejection.

Claim 20 is rejected for the following reasons: See Claim 6 rejection.

Claims 8, 9, 11, 13, 22, and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over. [A] in view ofOracleCorperation, "Oracle9i SQL Referance, Release 2 (9.2), known hereafter as [B]. 31. Claim 8 is rejected for the following reasons:

Reference [A] teaches the claims upon which claim 8 is dependant. The system inherently contains and SQL Engine and a server side stub, such as the code that facilitates the network communications. The admitted prior art also teaches the SQL engine and the Server side stub, see paras 14 and 15 of the background. Reference [B] teaches Aggregate functions, 4-6 to 4-8 that "return a single result based on groups of rows. "Thus, it would have been obvious to one of ordinary skill in the art to include aggregate functions in the system with the SQL engine and server side stub. As can be seen for the following reasons: First, the RETURNING clause is intended to eliminate the need for a select clause, [A] page 9 "This eliminates the need to SELECT the row after an insert or update, or before a delete. "Thus to truly eliminate the need for the SELECT clause it would need to integrate all the features of the SELECT clause, i.e. the ability to perform aggregate functions. Which are used to, as stated on pages 9 and 10 of reference [A], "As a result, fewer network round trips, less server CPU time, fewer eursers, and less server memory are required. "Reference [A] also shows the need to aggregate data further as it states "Now do computations involving name and new_sal. "

Claim 9 is rejected for the following reasons: Reference [A] teaches the claimsupon which claim 9 is dependent. The system also inherently has an SQL engine and a client interface. Reference [B] teaches Aggregate functions, 4-6 to 4-8 that "return a single result based on groups of rows. "Thus, it would have been obvious to one of ordinary skill in the art to include aggregate

functions in the system. As can be seen for the following reasons: First, the RETURNING clause is intended to eliminate the need for a select clause, [A] page 9 "This eliminates the need to SELECT the row after an insert or update, or before a delete. "Thus to truly eliminate the need for the SELECT clause it would need to integrate all the features of the SELECT clause, i.e. the ability to perform aggregate functions. To, as stated on pages 9 and !0 of reference [A], "As a result, fewer network round trips, less server CPU time, fewer cursers, and less server memory are required. "Reference [A] also shows the need to aggregate data further as it states "Now do computations involving name and new_sal. "

Claim 11 is rejected for the following reasons: Reference [A] teaches the claims upon which claim 11 is dependent. The system also inherently has an SQL engine and a client interface. [B] teaches Aggregate functions, 4-6 to 4-8 that "return a single result based on groups of rows" and multiple aggregate functions, 4-7 "AVG(MAX(SAL)). "Thus, it would have been obvious to one of ordinary skill in the art to include aggregate functions in the system. As can be seen for the following reasons: First, the RETURNING clause is intended to eliminate the need for a select clause, [A] page 9 "This eliminates the need to SELECT the row after an insert or update, or before a delete. "Thus to truly eliminate the need for the SELECT clause it would need to integrate all the features of the SELECT clause, i.e. the ability to perform .e aggregate functions. To, as stated on pages 9 and 10 of reference [A], "As a result, fewer network round trips, less server CPU time, fewer cursers, and less server memory are required. "Reference [A] also shows the need to aggregate data further as it states "Now do computations involving name and new_sal. "

Claim 13 is rejected for the following reasons:

Reference [A] teaches the claims upon which claim 13 is dependent, as well inherently containing the call interface as it includes a server and a network as stated on pages 9 and 1 O, "fewer network round trips, less server CPU time,. Reference [B] teaches Aggregate functions, 4-6 to 4-8 that "return a single result based on groups of rows" and multiple aggregate functions, 4-7 "A VG(MAX(SAL)). " Thus, it would have been obvious to one of ordinary skill in the art to include aggregate functions in the system. As can be seen for the following reasons: First, the RETURNING clause is intended to eliminate the need for a select clause, [A] page 9 "This. eliminates the need to SELECT the row after, an insert or update, or before a delete. "Thus to truly eliminate the need for the SELECT clause it would need to integrate all the features of the SELECT clause, i.e. the ability to perform aggregate functions. To, as stated on pages 9 and 10 of reference [A], "As a result, fewer network round trips, less server CPU time, fewer cursers, and less server memory are required. "Reference [A] also shows the need to aggregate data further as it states "Now do computations involving name and new_sal. "

Claim 22 is rejected for the following reasons: See claim 11 rejection.

Claim 24 is rejected for the following reasons: See claim 13 rejection.

Claims 12 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over [A] in view of [B] in further view of US patent Number 6567803 know hereafter as Ramasey.

Claim 12 is rejected for the following reasons:

Reference [A] teaches the claims upon which claim 11 is dependent. The system also inherently has an SQL engine and a client interface. Reference [B] teaches Aggregate functions, 4-6 to 4-8 that "return a single result based on groups of rows" and multiple aggregate functions, 4-7 "A VG(MAX(SAL)). "See Claim 11 rejection for more information. Ramasay teaches using

operator trees corresponding to different aggregate functions (Col 4 lines 6-16) and an access plan that lists function and includes structures pointing to workspaces performing the functions(Col 4 lines 17-46). It would have been obvious to one of ordinary skill in the art at the time of the invention to include these features as they provide a method for providing an optimized query that saves processing time and allow for parallel processing.

Claim 23 is rejected for the following reasons: See claim 12 rejection.

Claims 14 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over [A] in view of applicants admitted prior art.

[A] teaches the claims upon which claims 14 and 25 are dependant, however it fails to expressly disclose performing the aggregations on old values. This is taught in the second and third paragraphs of the instant applications background. Thus it would have been obvious to one of ordinary skill in the art at the time of the invention to include these features, as it would provide the user with useful information. Response to Arguments.

(10) Response to Argument

Response to Argument A

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Appellant has requested review of "Claims 1-7, 10, and 15-20 have been rejected under 35 U.S.C. § 103(a) as being allegedly

unpatentable over PL/SQL User's Guide and Reference Release 2(9.2) ('reference [A]') *in view of Applicant admitted prior art ('AAPA')*. Reference [A] includes: (a) pages 51- 55 of Chapter 5, entitled 'PL/SQL Collections and Records'; and (b) pages 1-13 of Chapter 12, entitled 'Tuning PL/SQL Applications'." (Emphasis added) Page 8 of the instant appeal brief. However, it must first be noted that appellant does not even acknowledge the AAPA in any his arguments.

Appellant's first argument is that the prior art of record fails to teach or suggest all the features of claims 1 and 15.

[A] Teaches, receiving a database statement that specifies a DML operation that modifies data in one or more columns in a database,(Page 53 of chapter 5, which teaches modifying a "sal" column, this is written in SQL which is a DML) and contains a clause that specifies an aggregate operation to be performed on a plurality of values associated with the data(Page 53 of chapter 5, teaches aggregating that values of the ename, jab, and sal variables into an emp_info variable); and In response to receiving the database statement, performing: the DML operation on the one or more columns in the database. performing the aggregate operation on the plurality of values(the query is inherently run and thus performing these features) and returning as a result of the database statement a result of the aggregate operation(Page 53 of chapter 5, teaches aggregating that values of the ename, jab, and sal variables into an emp_info variable which is then returned). [A] However, fails to expressly disclose the use of an aggregate function for multiple rows in a returning clause.

Yet, [A] does suggest that RETURNING clause is intended to eliminate the need for a select statement, [A] page 9 "This eliminates the need to SELECT the row after an insert or update, or before a delete. " Which results in "fewer network round trips, less server CPU time, fewer cursers, and less server memory are required. " (pages 9 and 10 of reference [A])

SELECT statements have the ability to perform specified aggregate functions across multiple rows. This is evidenced by chapter 6 page 2 of [A] :

" PL/SQL lets you use all the SQL functions including the following aggregate functions, which summarize entire columns of Oracle data: AVG, COUNT, GROUPING, MAX, MIN, STDDEV, SUM, and VARIANCE. Except for COUNT(*), all aggregate functions ignore nulls...

SELECT COUNT(DISTINCT job) INTO job_count FROM emp;"

Appellant argues that there would be no motivation to provide support for aggregates functions that are used in SELECT statements in a RETURNING clause. However, one of ordinary skill in the art at the time of the invention would have been motivated to do so to eliminate the need to have to run a SELECT statement after and insert or update of before a delete in which he wanted to perform an aggregate on a column, and thus have fewer network round trips, less server CPU time, fewer cursers, and less server memory are required. Also applicant states in the Background that programmers were using the iterative technique to perform these aggregate functions on the client-side before the RETURNING clause had the

ability to perform multi-row aggregates on the server side. Which further shows that users wished to perform these functions that they had been allowed to use in SELECT statements.

Furthermore, the present case satisfies the rational of combining prior art elements according to known methods to yield predictable results. MPEP 2143[R-6] A ("Exemplary rationales that may support a conclusion of obviousness include: (A) Combining prior art elements according to known methods to yield predictable results;"). The Prior art teaches RETURNING clauses, and that it was known how to program and provide for multi-row aggregates on the server side in Select statements. The AAPA shows that one of ordinary skill in the art knew how to program multi row aggregates on the client side.

> " Unfortunately, the conventional implementation of the returning clause only returns a single value-of-interest. Therefore, programmers must resort to more complex techniques in situations where (1) a first operation changes many values, thereby creating many values-of- interest, and (2) an aggregate operation is to be performed on the values-of-interest associated with the first operation. One such technique is referred to herein as the iterative technique.
>
> According to the iterative technique, the first operation (which changes many values) is performed by iteratively executing code that only changes a single value. During each iteration, the returning clause is used to return the value-of-interest associated with that iteration. For example, assume that a user wants to increment several values in C1, and to keep track of all of the new, post-update values that are changed during the update operation. Such an operation could be performed using the following code segment. ..."

Thus, the difference between the prior art RETURING clause and the instant claim is the claimed RETURING clause as claimed can perform multi-row aggregates, but given that programmers were performing this functionality on the client side and it was known how to perform multi row aggregate on the server side (as was done in SELECT statements) this combination would have been obvious as, the combination of these known methods would yield a RETURNING clause in which one could specify multi-row aggregates to provide the predicable result of having the ability to perform aggregate functions on values in the RETURNING clause. Thus, even if there is no motivation the rejection is still proper.

Appellant's second contention is that [A] teaches away. However, Appellant fails to recognize the distinction between not teaching a limitation and teaching away. [A] does state that you can only use the RETURNING clause when operating on one row, yet goes not go as far as to say that one would not or could not program functionality of acting on multiple rows into the returning clause or that one would not want to perform an aggregate function in the RETURNING clause. This instead statement in [A] appears to be a warning to users who would likely attempt to act on multiple rows expecting the RETURING clause to have the ability to act on multiple rows. This argument also fails as even if [A] were considered to teach away it would be outweighed by the suggestive power of the AAPA. MPEP 2143.01[R-6] II ("The test for obviousness is what the combined teachings of the references would have suggested to one of ordinary skill in the art, and all teachings in the prior art must be considered to the extent that they are in analogous arts. Where the teachings of two or more prior art references conflict, the examiner must weigh the power of each reference to suggest solutions to one of ordinary skill in

the art, considering the degree to which one reference might accurately discredit another.). As the AAPA shows how programmers were performing the aggregate operations with RETURNING clauses that only operated on single rows.

Appellant's third contention is that the reference fails to teach of suggest integrating all features of the SELECT statement. First, a teaching, suggestion, and motivation test would only require a motivation to add in the aggregate function. Second, motivation is not required as the present case satisfies the rational combining prior art element according to known methods to yield predictable results. MPEP 2143[R-6] A. Third, eliminating the need for a SELECT statement does suggest incorporation of the elements of a SELECT statement. Last, Appellant once again falls prey to the fact that he has failed to consider the references in combination, as the AAPA shows the workaround that was being used by programmer because the aggregate function capability of the SELECT statement was not present in the RETURNING clause.

Appellant's final argument with regards to claims 1 and 15 is that [A] fails to show the need to aggregate the information further. Although applicant provides no support for why failing to show the need to further aggregate data would invalidate the 35 U.S.C. 103 rejections, and the examiner concedes that the statement "now do computations involving name and new_sal" does not expressly disclose doing aggregates on the data, however, the AAPA shows the computations including aggregating the data.

Thus, given that appellant failed to argue the references in combination, there was motivation to combine, all elements are taught in the prior art, and, even if there was no motivation, the rejection is still a combination of known methods to yield a predictable result the examiner should be affirmed.

Response to argument B

This argument relies on claims 1 and 15 being allowable in light of argument A. Thus, as the examiner should be affirmed for the rejection under 35 U.S.C. 103 above as it was proper, he should also be affirmed with regards to argument B.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Cory Bell

CA

Conferees:

Charles Rones

Mohammad Ali

HOSAIN ALAM
SUPERVISORY PATENT EXAMINER

CHARLES RONES
SUPERVISORY PATENT EXAMINER